(12) UK Patent Application (19) GB (11) 2 386 442 (13) A

(43) Date of A Publication 17.09.2003

(21) Application No 0205817.0

(22) Date of Filing 12.03.2002

(71) Applicant(s)
Toshiba Research Europe Limited
(Incorporated in the United Kingdom)
32 Queens Square, BRISTOL, BS1 4ND,
United Kingdom

(72) Inventor(s)
Jonathan David Lewis

(74) Agent and/or Address for Service
Marks & Clerk
57-60 Lincoln's Inn Fields, LONDON,
WC2A 3LS, United Kingdom

(51) INT CL^7
G06F 9/46

(52) UK CL (Edition V)
G4A AFN

(56) Documents Cited
None

(58) Field of Search
UK CL (Edition T) G4A AFN
INT CL^7 G06F 9/46 15/16 15/177 17/30
Other: Online: WPI, EPODOC, JAPIO, INSPEC

(54) Abstract Title
Allocation of data into time frames and allocation of particular time frame data to a particular processor

(57) A data processing system 10 comprising a plurality of processors, a first controller that allocates data to one of a plurality of time frames and a second controller that allocates data associated with a particular time frame to a particular processor for processing, with the aim of minimising data transfer between processors. The processors may be connected via a matrix 13 to re-configurable logic blocks or accelerators 11, and preferably there are twice as many accelerators as processors. The processors may be DSPs, ASICs or ASSPs. Preferably the system, and associated method, are for use in a mobile communications terminal such as a mobile phone where data via received in air interface timeslots.

FIGURE 1

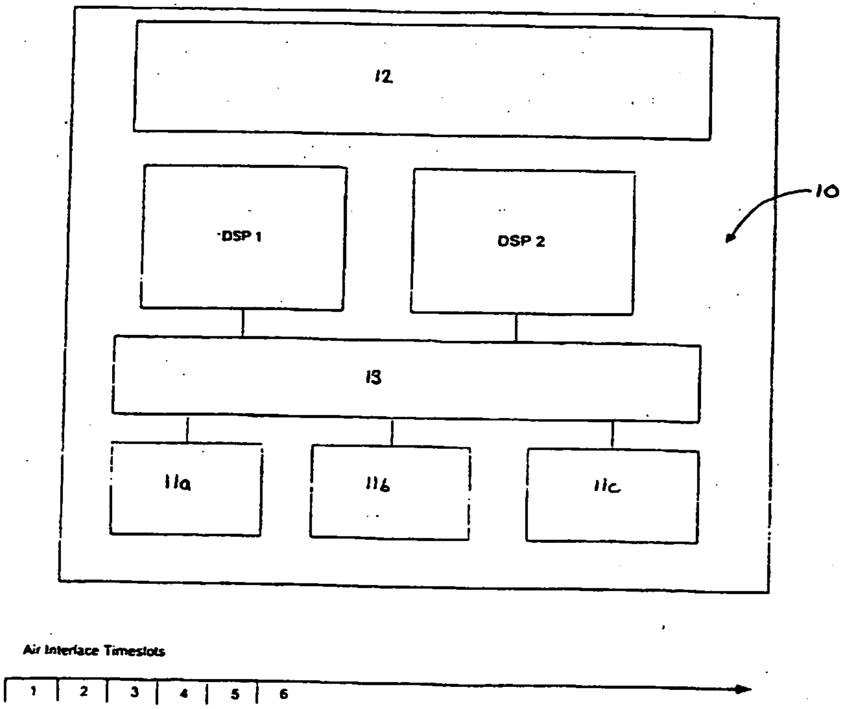
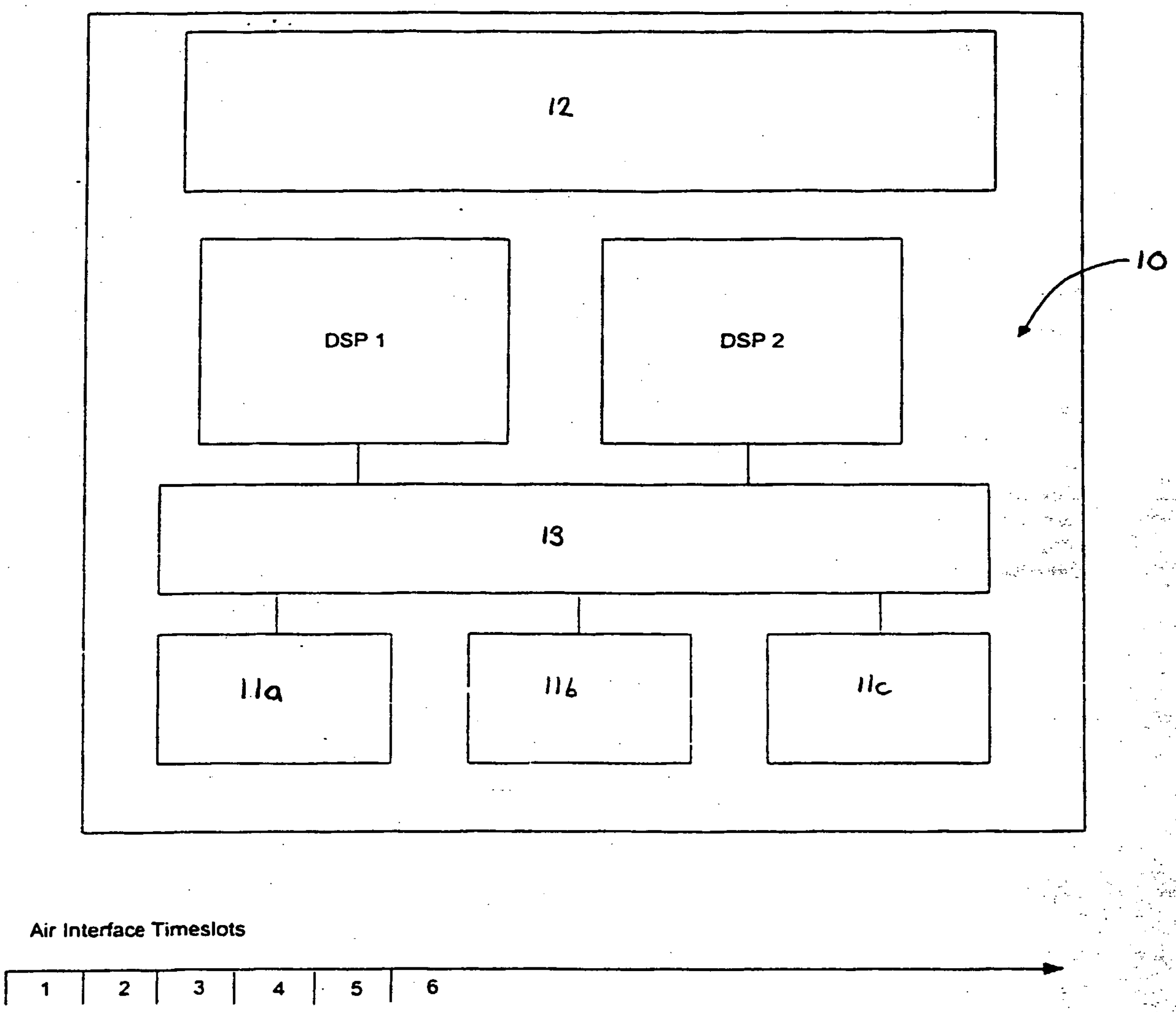


FIGURE 1


12
DSP 1
DSP 2
10
13
11a
11b
11c
Air Interface Timeslots
1
2
3
4
5
6

2386442

Allocation of Hardware Accelerators

The present invention relates to systems and methods in the field of signal and data processing and particularly controllers in multi-processor systems. Even more particularly the invention relates to multi-processor system control in a mobile telecommunications device.

With the onset of third generation (3G) telecommunication standards, there is a need for the complexity and configurability of signal processing equipment to improve and adapt. The existing second generation (2G) cellular standards and intermediate standards (2.5G), such as GSM, D-AMPS and Narrowband CDMA are directed to the delivery of speech and low bit-rate data services. The need to support wireless broadband multimedia services over 3G systems, such as Wideband CDMA, therefore requires increased signal processing power in comparison, in order to support the higher data rates and quality-of-service levels. This also applies to multi-band devices.

Various Digital Signal Processor (DSP) arrangements have been devised with a view to improving performance, such as parallel execution via deep pipelining and multiple execution units. DSPs designed for parallel processing having multiple high-speed data and memory buses, a number of I/O interfaces and on-chip controllers for inter-processor communication and have instruction sets that rapidly execute instructions. Key to achieving enhanced real-time signal processing performance in telecommunications equipment, such as digital receivers is to harness the power of DSPs in the most effective manner by achieving optimum processor-to-processor communication throughput. For example, it is desirable to minimise the amount of data that is moved between processing elements.

In this regard, in the operation of multi-processor systems, typically each DSP is assigned a particular task, so that one DSP performs a particular function on data and then passes the data to another DSP for subsequent processing. The first DSP then performs the particular function on another set of data before passing it on to the second DSP and so on. This is not a desirable arrangement, as a large amount of data is passed between the processing elements.

There is therefore a need for an improved processing arrangement.

Reconfigurable devices, such as field programmable gate arrays (FPGAs), are a compromise between a pure software solution and a pure hardware solution. Reconfigurable devices are digital circuits that can be programmed by reconfigurable logic in order to dynamically create and modify custom digital circuits. This ability to create and modify digital logic without physically altering the hardware provides a more flexible and lower cost solution to the implementation of custom hardware. Reconfigurable logic exploits program parallelism as programming is accomplished by mapping algorithms on demand to a pool of FPGAs. This approach of an FPGA assuming the logic design required to implement the algorithm is to be contrasted to that of a programmed processor which executes a sequence of instructions on predefined hardware resources.

Because the function of the reconfigurable logic device is defined by software, design errors can be corrected without having to fabricate new hardware. Existing system hardware may also be modified and upgraded without any physical modifications. Only a change to the software used by the reconfigurable logic device is required. Reconfigurable devices therefore offer an increased benefit in computational density over microprocessors, and for highly regular computations, reconfigurable architectures are generally superior to traditional processor architectures. However, on tasks with high functional diversity, microprocessors use custom hardware more efficiently than reconfigurable devices. Hence a combination of the two may be utilised.

While reconfigurable devices have proven extremely efficient for processing tasks, there is still scope for further improvement in their efficiency. For example, in present systems when a particular instruction for a routine is received for which the architecture is not configured, a reconfigurable device must be reconfigured, which takes time out of the overall signal processing procedure, particularly where one or more other routines are dependent upon the outcome of that particular routine.

Applications such as 3G cellular handsets and associated hardware, including base stations, can require a greater performance level, particularly in terms of MIPS, that a conventional DSP core cannot deliver. Hence, in a computing device, the time taken to configure the reconfigurable device may be significant in terms of multiple clock cycles, so the present invention seeks to address this problem.

There is hence also a need for a signal/data processing system that is capable of improved performance. In particular, this need is in relation to a low-energy wireless device.

There is also a need to overcome or alleviate at least one of the problems of the prior art.

In one aspect the present invention provides a data processing system comprising a plurality of processors; a plurality of processors; a first controller adapted to allocate data to be processed to one of a plurality of time frames; and a second controller adapted to allocate data associated with a particular time frame to a particular processor to process.

In a related aspect, the present invention also provides a method of processing data in a data processing system comprising receiving data and dividing it into a plurality of time frames; and allocating data associated with a particular time frame to a particular processor to process.

These aspects of the invention utilises the predictability, inherent in a time slot or time frame arrangement, of when a particular block configuration is required to maximise processor efficiency.

The present invention will now be described with reference to the following non-limiting preferred embodiments in which:

Figure 1 illustrates a schematic architecture for implementing the present invention.

With reference to Figure 1 a schematic processor arrangement 10 is illustrated which includes a controller 12 for overseeing the distribution of data throughout the processor, as well as a number of Digital Signal Processors, being, in this example, DSP 1 and DSP 2. Although only two DSPs are illustrated in Figure 1, any number may be utilised.

The DSPs are associated with a number of reconfigurable logic blocks (RLBs). These reconfigurable logic blocks (11a, 11b, 11c) may be hardware accelerators or alternatively, a reconfigurable portion of a hardware accelerator, such as a row of a hardware accelerator. Although only three RLBs are illustrated in Figure 1, again, any number may be utilised. In Figure 1, the DSPs are illustrated as being connected to the RLBs via a connection matrix or bus 13. This is one connection arrangement and other alternatives are possible. It is to be appreciated that Figure 1 in general is intended to show that a number of DSPs and RLBs are used and may be connected in a variety of different manners. The RLBs are preferably fully shared and hence available for use by all the DSPs in the multi-processor environment.

When performing a given function, the DSPs may require the use of one or more of the RLBs in order to accelerate the operation. In accordance with a first embodiment of the present invention, the assignment of data to the plurality of digital signal processors in the multi-processor system is based upon a timeslot/frame structure. One or more sets of data are allocated to a single frame or timeslot and all processing for a given timeslot is undertaken by an individual DSP. Each processor can be configured to perform a number of sequential tasks, which reduces the need to move large amounts of data between processors.

Hence, considering the Figure 1 arrangement, with two digital signal processors, DSP1 and DSP2, DSP1 may be allocated one or more sets of data associated with timeslot 1 for processing and DSP2 could be allocated one or more sets of data associated with timeslot 2 for processing. However, it is to be noted that since the scheduling is time-

The allocation decisions could then, in one embodiment of the invention, be based upon priorities in terms of the time slot priority. For example, time slot priority could be based upon time slot ordering such that a time slot with a low order value (e.g. time slot 1) could have a higher priority than a timeslot with a higher order value (e.g. time slot 4).

In particular, consider that the DSP processing allocated data from a timeslot with a high priority value has been allocated two RLBs to use. One of those RLBs has not yet been used by that DSP and that RLB has a particular configuration. If that particular configuration is subsequently required by a DSP processing data associated with a time slot with a lower priority value, the controller could determine whether the particular configuration of the RLB is specifically required by the DSP operating in the higher priority value time slot. If, for instance, the RLB was randomly allocated to the DSP operating in the high priority time slot, it would be more efficient to reallocate that RLB to the DSP operating in the lower priority time slot and allocate another free RLB to the higher priority time slot DSP. If, however, it is known that the particular configuration of the RLB is required by the higher priority time slot DSP, then the RLB would be not reallocated.

On the other hand, when allocating blocks to lower priority timeslot DSPs, a configured block that a higher priority timeslot DSP has not yet used should not be allocated. Similarly, if it is known that a particular configuration of a block is required by a higher priority time slot DSP, then that block should also not be allocated to a lower priority timeslot DSP.

In a still alternative embodiment of the invention, if a DSP operating in a timeslot of high priority value requires an RLB and a free one is not available, it is given a greater priority than DSPs operating in low priority time slots. In view of this greater priority, the high priority timeslot DSP takes a block from a DSP operating in a timeslot of lower priority value. The determination of which block to take could be based upon the relative priority value of the time slot in which the RLB has been already allocated and/or the function of the block. For example, the high priority time slot DSP could take a block <u>of the same function</u> from a DSP operating in a timeslot with a lower priority value. If a block of the same function is not available, then it could take one from a DSP operating in a timeslot with a higher priority value and reconfigure it. Preferably the RLB is taken from the DSP operating in a time slot with the lowest priority value, particularly where the determination is not based upon the function of the block.

Where a block is taken from a low priority timeslot DSP and that DSP had already initiated a function in that block, preferably the task is halted and the context of the initial task saved so that the DSP can continue from where it had been interrupted. In this regard, the task could be continued in the same block, once the DSP operating in the higher priority value timeslot has finished with it, or in a wholly different block.

In another embodiment of the invention, each RLB has a timestamp to show when it was last used. This time stamp can then be utilised in the allocation of the RLBs. For example, a store of the timestamps could be provided, and if reconfiguration is required, then a time store access means would access the store to determine which free block has not been used for the longest time. That free block could then be reconfigured.

According to another embodiment of the invention, a look-ahead mechanism is utilised so that, if pre-configuration is required, it occurs before the RLB is required. For example, a controller may, when a particular instruction is being executed by a particular processor, look a number of instructions ahead to see what configurations are to be shortly required.

Preferably, the system has twice the number of reconfigurable blocks as processors. In this way, for each DSP, there would be one active reconfigurable block and one being configured, where necessary, for the next processing task.

Alternatively, blocks could be reconfigured before the actual program thread that requires them, runs the blocks. For example, reconfiguration, where necessary, could occur in the time slot preceding the timeslot in which the program is to run. This embodiment of the invention is particularly applicable to systems where the blocks are fully shared.

based, DSP1 would in effect have two timeslots to complete its processing, as it would not be required to process any additional data until time slot 3. The same would apply to DSP2, which would not be required to process any additional data until timeslot 4. As each processor is configured to perform a number of sequential tasks, data is not moved between processors during the execution of those sequential tasks, only between the processor concerned and the relevant hardware accelerator.

It should also be possible with suitable scheduling based around the timeslot structure, to reduce the time when a processor cannot get access to a required hardware accelerator.

It is to be appreciated that this embodiment of the invention may be applied to any number of processors. For example, where four DSPs are used, the timeslots may be allocated in any manner between the DSPs. For instance, each set of four time slots may be allocated sequentially to the four DSPs or in any other multiples thereof.

It is also to be appreciated that while this invention is particular applicable to signal processors that receive data via a TDMA system, where signals are transmitted across the air interface in time slots, this invention need not be limited to use in a TDMA communication system. In this regard, where the signal processing system in accordance with the invention is used in another type of communication system, instructions received by the mobile device, such as via CDMA, would be received by a RAKE receiver and appropriately demodulated. The received data could then be allocated into a frame structure in order to utilise the present invention.

Another embodiment of the invention is directed to the efficient allocation of RLB resources, particularly with a view to overcoming the time normally taken to reconfigure RLBs and ensuring that a block is available to a processor when required. In this regard, each RLB may be marked so as to indicate its current configuration as well as with an indication of the status of the block such as whether it is free or busy. These indications may be provided, for example, by various storage means, such as via flags associated with the blocks or via updateable look-up tables or lists. Hence, as one processor requires a RLB for processing, it takes ownership of a particular module until the task is completed, marking the free/busy flag as "busy". When it has finished the block is marked as free. The current configuration marker is preferably one or two bits that reference predefined operations.

A controller utilises these markers in order to provide a systematic allocation of free RLBs. In operation, the controller would be notified that a DSP wants a particular function. The controller would then check all free blocks to determine if that function is already configured in a free block. If so, then the DSP is allocated the pre-configured block. Alternatively, if no free block is appropriately configured, then any free block is reconfigured and allocated to the DSP. In this way, reconfigurable blocks are more likely to be available when required and without the need for reconfiguration.

While it is preferable that only the configuration of free blocks are checked, that is, blocks that are not being used, it is within the scope of this invention to check the configuration of all reconfigurable logic blocks.

Where specialised functions are defined in the RLB, one marker reference of the current configuration marker may relate generally to such operations. In effect, as only one marker would relate to a plurality of specialised functions, blocks with this marker would not be able to be reused, as the exact operation would not be determinable simply by analysing the current configuration marker. Therefore, where such a marker is used, the controller could allocate such free blocks for reconfiguration before all other blocks. In this way the efficiency of re-allocating blocks can be improved.

In another embodiment of the invention, the controller decides which block to assign to each DSP based upon knowledge of reconfiguration time and current thread of each device. For example, where four DSPs exist in the multi-processor system, the controller could keep a list, in a storage means, of all the configuration functions required by the respective DSPs during the first cycle of timeslots for the four DSPs. This list could be updateable after each processor completes processing the one or more sets of data associated with its designated timeslot.

Alternatively, the controller may predict the next processing task and reconfigure blocks, where required, according to the next predicted function or based upon history. For example, a given block could be automatically reconfigured according to the previous transition. That is, each task could have an associated "most likely" transition field. This could be the last transition that was made from the particular task. Alternatively, the controller may have a store of task sequences that are likely to occur and pre-configure a block based upon a particular sequence. That is, the record may indicate that where task A followed by task B occurs, task C is likely to next occur.

Any combination of these embodiments is within the scope of this invention.

The techniques and arrangements of the present invention assist in maximising the efficiency of signal processors and hence assist in the reduction of power consumption, silicon size as well as cost. Signal processors embodying the present invention may therefore be employed in a mobile terminal, such as the chipset of a multimode mobile handset. Alternatively the signal processors could be employed in a base station, and may be embodied in a semiconductor, hardware, or software, or a combination thereof.

Variations and additions are possible within the general inventive concept as will be apparent to those skilled in the art. For example, it is not essential to the invention that DSPs are utilised. Alternative processors may be used, such as Application Specific Integrated Circuits (ASICs) or Application Specific Standard Products (ASSPs), where suitable.

It will be appreciated that the broad inventive concept of the present invention may be applied to any field utilising reconfigurable computing, and the embodiments shown are intended to be merely illustrative and not limiting. For example the present invention may be utilised to enhance signal/data processing in areas such as encryption/decryption, compression, pattern and string matching, sorting, physical system simulation, video and image processing and specialized arithmetic.

CLAIMS:

1. A data processing system comprising:
a plurality of processors;
a first controller adapted to allocate data to be processed to one of a plurality of time frames; and
a second controller adapted to allocate data associated with a particular time frame to a particular processor to process.

2. The processing system of claim 1 wherein the plurality of time frames and associated data are allocated cyclically to the processors.

3. The processing system according to claim 1 or 2 further comprising:
one or more reconfigurable logic blocks in communicable relation with the processors; and
a control unit for controlling configuration of the reconfigurable logic blocks during processing.

4. The processing system according to claim 3 wherein the control unit is adapted to:
receive information relating to a required logic block configuration for carrying out a desired task;
check the current configuration of the one or more reconfigurable logic blocks; and
where a reconfigurable logic blocks matches the required function, allocating that reconfigurable logic block to carry out the desired task.

5. The processing system of claim 3 or 4 further comprising a free/busy indicator for each reconfigurable logic block.

6. The processing system according to any one of claims 3 to 5 further comprising a priority storage means for storing a list of time frame priority values, such that

a time frame with a high priority value has a higher priority for the allocation of reconfigurable logic blocks than a time frame with a low priority value.

7. The processing system of claim 6 further comprising:
a block storage means for storing a list of reconfigurable logic blocks allocated to a processor during a particular time frame; and
a logic block allocator which utilises the list in the block storage means such that any reconfigurable logic block with a particular configuration that has been allocated to a first processor allotted data from a time frame with a low priority value and which is subsequently required by a second processor processing data from a time frame with a high priority value, will be reallocated to the second processor.

8. The processing system according to claim 6 or 7 wherein the time frame priority values are numerical values allocated to each time frame sequentially on an ascending scale.

9. The processing system according to any one of claims 3 to 8 further comprising a predictor for predicting the next logic block configuration to be required by a processor.

10. The processing system according to claim 9 wherein the predictor comprises:
storage means for storing a list of known configurations and associated subsequent configurations; and
a comparator for comparing the current configuration of a reconfigurable logic block utilised by the processor with the list of known configurations, and where the current configuration matches an entry in the list of known configurations, the subsequent configuration associated with the matched entry is determined as the next configuration to be required by the processor.

11. The processing system of claim 9 wherein the predictor comprises:

sequence storage means for storing a list of configuration sequences and the corresponding next configurations associated with each configuration sequence; and
a comparator for comparing a sequential list of previous configurations used by the processor with the list of configuration sequences in order to determine the next configuration to be required by the processor.

12. The processing system according to any one of claims 3 to 11 further including a timestamp store for storing information indicating the period of time each of the one or more reconfigurable logic blocks have not been utilised; and
timestamp store access means for determining the reconfigurable logic block which has not been utilised for the longest time, such that if reconfiguration of a logic block is required by a processor, the determined block is allocated.

13. A method of processing data in a data processing system comprising:
receiving data and dividing it into a plurality of time frames; and
allocating data associated with a particular time frame to a particular processor to process.

14. The method of claim 13 wherein the data in each of said plurality of time frames is allocated cyclically to a plurality of said processors.

15. The method of processing data in a data processing system according to claim 13 or 14, wherein the data processing system further comprises one or more reconfigurable logic blocks associated with the plurality of processors, the method further comprising the steps of:

a) determining the next configuration of a reconfigurable logic block to be required by a particular processor to process its data;
b) checking the current configuration of the one or more reconfigurable logic blocks, and:
    (i) where the current configuration of one of the reconfigurable logic blocks matches the required configuration, allocating the use of

that reconfigurable logic block for implementing the required configuration; or

(ii) where none of the existing configurations of the one or more reconfigurable logic blocks matches the required configuration, configuring one of the reconfigurable logic blocks for implementing the required configuration.

16. The method of claim 15 wherein the current configuration of only the reconfigurable logic blocks which are not being used is checked.

17. The method according to claim 15 or 16 further comprising the steps of:
looking ahead in a sequence of instructions to determine if a particular configuration of a logic block is to be required shortly; and
where a particular configuration is to be required shortly, commencing step (b) before the particular configuration is required.

18. The method according to claim 15, 16 or 17 further comprising the steps of:
determining the current configuration of a reconfigurable logic block;
comparing the current configuration with a list of known configurations, and where the current configuration matches an entry in the list of known configurations, the subsequent configuration associated with the matched entry is determined as the next configuration to be required by the processor.

19. The method of claim 15, 16 or 17 further comprising the steps of:
determining a sequential list of previous configurations used by a particular processor;
comparing the sequential list with a predetermined list of configuration sequences, and where the sequential list matches an entry in the list of configuration sequences, a subsequent configuration associated with the matched entry is determined as the next configuration to be required by the processor.

20. The method according to any one of claims 15 to 19 further comprising the steps of:
allocating a priority value to each time frame; and
allocating a reconfigurable logic block to a processor based upon the priority value of the time frame associated with the data allocated to the processor.

21. The method of claim 20 wherein the priority value is determined by the ordering of the time frames.

22. The method of claim 20 or 21 wherein if a first processor processing data associated with a time frame of high priority value requires a logic block of a particular configuration already allocated to a second processor allocated data associated with a time frame of low priority value, then the block is reallocated to the first processor.

23. The method according to claim 22 wherein the first processor will reallocate a logic block to the second processor where the logic block was randomly allocated to the first processor and the specific configuration of the logic block is required by the second processor.

24. A controller for use in a mobile communication device for performing a method according to any one of claims 14 to 23.

25. A mobile communication terminal comprising a data processor according to any one of claims 1 to 12.

26. A mobile communication base station comprising a data processor according to any one of claims 1 to 12.

27. A data processor substantially as herein described with reference to the accompanying drawing.

CHARTER MARK

PATENTS • DESIGNS • The Patent Office • COPYRIGHT • TRADE MARKS

INVESTOR IN PEOPLE

Application No:	GB 0205817.0	**Examiner:**	Richard Baines
Claims searched:	1 - 27	**Date of search:**	24 September 2002

Patents Act 1977
Search Report under Section 17

Databases searched:

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.T): G4A (AFN)

Int Cl (Ed.7): G06F 9/46, 15/16, 15/177, 17/30

Other: Online: keywords in EPODOC, WPI, JAPIO, INSPEC

Documents considered to be relevant:

Category	Identity of document and relevant passage	Relevant to claims
	NONE	

X	Document indicating lack of novelty or inventive step	A	Document indicating technological background and/or state of the art.
Y	Document indicating lack of inventive step if combined with one or more other documents of same category.	P	Document published on or after the declared priority date but before the filing date of this invention.
		E	Patent document published on or after, but with priority date earlier than, the filing date of this application.
&	Member of the same patent family		